Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Previously Presented) A processing core, comprising:

a first source register including a plurality of first operands;

a plurality of second operands, wherein:

the plurality of second operands are equal in value to an immediate value, and

the immediate value is specified in an instruction that identifies the first source register;

a bitwise inverter coupled to at least one of the first plurality of operands and the second plurality of operands;

a destination register including a plurality of results;

a plurality of arithmetic processors respectively coupled to the first operands, second operands and results, wherein each arithmetic processor computes one of a sum and a difference of the first operand and a respective second operand.

2. (Original) The processing core of claim 1, further comprising an integrated circuit which includes the first source register, destination register and arithmetic processor.

3. (Canceled)

4. (Currently Amended) The processing core of claim 1, wherein:

each arithmetic processor computes at least one of:

the result of the first operand plus another operand plus the immediate value; and

the result of the first operand minus another operand minus the immediate value**; and**

each of the first operand, the another operand and the immediate value are represented with a plurality of bits.

5. (Previously Presented) The processing core of claim 1, wherein the immediate value is signed.

6. (Previously Presented) The processing core of claim 1, further comprising a prescaler which scales the immediate value.

7. (Original) The processing core of claim 1, wherein a first width of the first source register is a positive integer multiple of a second width of the first operand.

8. (Previously Presented) The processing core of claim 1, wherein the sum and the difference are performed on a same carry look-ahead adder.

9. (Previously Presented) A method for performing arithmetic processing, the method comprising the steps of:

loading a first and second operands from a primary source register;

loading a third and fourth operands, wherein:

the third and fourth operands are an immediate value specified in an instruction, and

the third and fourth operands are equal in value;

scaling the third and fourth operands according to a predetermined scaling factor;

performing an arithmetic function on the first and third operands to produce a first result;

performing the arithmetic function on the second and fourth operands to produce a second result; and

storing the first and second results in a destination register.

10. (Original) The method for performing arithmetic processing of claim 9, further comprising a step of inverting the third and fourth operands.

11. (Original) The method for performing arithmetic processing of claim 9, further comprising a step of adjusting at least one of the first and second results to avoid saturation of the destination register.

12. (Currently Amended) The method for performing arithmetic processing of claim 9, wherein the step of performing an arithmetic function on the first and third operands comprises calculating the first operand plus the ~~second~~ **third** operand plus a positive integer**, wherein each of the first operand, the third operand and the positive integer are each represented with a plurality of bits**.

13. (Currently Amended) The method for performing arithmetic processing of claim 9, wherein the step of performing an arithmetic function on the second and fourth operands comprises calculating the second operand minus the fourth operand minus a positive integer**, wherein each of the first operand, the third operand and the positive integer are each represented with a plurality of bits**.

14. (Canceled)

15. (Original) The method for performing arithmetic processing of claim 9, wherein the predetermined scaling factor is divisible by two.

16. (Original) The method for performing arithmetic processing of claim 9, wherein the two performing steps are performed, at least partially, coextensive in time.

17. (Previously Presented) The method for performing arithmetic processing of claim 9, wherein the two performing steps use a ripple look-ahead adder.

18. (Previously Presented) A method for performing arithmetic processing, comprising the steps of:

loading a first and second operands from a primary source register;

loading an immediate value that has a value expressly specified in an instruction;

performing an arithmetic function on the first operand and immediate value to produce a first result;

performing the arithmetic function on the second operand and immediate value to produce a second result; and

storing the first and second results in a destination register.

19. (Currently Amended) The method for performing arithmetic processing of claim 18, wherein the immediate value ~~has a width~~ **is comprised** of nine bits.

20. (Currently Amended) The method for performing arithmetic processing of claim 18, wherein the immediate value ~~has a width~~ **is comprised** of thirteen bits.

21. (Original) The method for performing arithmetic processing of claim 18, wherein the two performing steps are performed, at least partially, coextensive in time.

22. (Original) The method for performing arithmetic processing of claim 18, further comprising a step of adjusting at least one of the first and second results to avoid saturation of the destination register.

23. (Currently Amended) The processing core of claim 1, wherein the instruction is a **very long instruction word (VLIW)** ~~VLIW~~ instruction.

24. (Previously Presented) The method for performing arithmetic processing of claim 9, wherein the steps of the method are initiated by a single instruction issue.